Boston University Computer Science Technical Report No. 2000-24, Dec. 2000 (revised Apr. 2001).

Region Segmentation via Deformable Model-Guided Split and Merge

Lifeng Liu and Stan Sclaroff*
Computer Science Department, Boston University
111 Cummington St., Boston, MA 02215
liulf@cs.bu.edu, sclaroff@cs.bu.edu

Abstract

An improved method for deformable shape-based image segmentation is described. Image regions are merged together and/or split apart, based on their agreement with an a priori distribution on the global deformation parameters for a shape template. Perceptually-motivated criteria are used to determine where/how to split regions, based on the local shape properties of the region group's bounding contour. A globally consistent interpretation is determined in part by the minimum description length principle. Experiments show that model-guided split and merge yields a significant improvement in segmention over a method that uses merging alone.

1 Introduction

Retrieval by shape is a key topic in content-based image retrieval research. Unfortunately, retrieval by shape requires object detection and segmentation. In [16, 23] we described a system for automatic shape-based indexing of image databases that employs a shape model to guide region grouping. A limitation of the system was that it employed model-based region grouping only, without region splitting. However, it is well-known that split/merge in region segmentation [11] tends to yield improved accuracy over methods that employ only splitting or only merging.

In this paper, a method is proposed for incorporating a model splitting step in the previous model-guided region grouping algorithm [16, 23]. Candidate region splits are determined using criteria that are motivated by human psychophysics. In addition, cues obtained from a statistical shape model are used to prioritize the selection of candidate region splits. The resulting algorithm combines bottom up processing with top-down (model-guided) processing. As will be seen in the experiments, the split and merge approach yields a significant improvement over the previous algorithm in detection and segmentation of deformable shapes. The method can detect multiple shapes even in the presence of poor initial segmentation, shadows, or when shapes in the image touch.

*This work was supported in part through Office of Naval Research Young Investigator Award N00014-96-1-0661, and National Science Foundation grants IIS-9624168 and EIA-9623865.

2 Related work

Some traditional methods for region splitting include separation of touching features by erosion/dilation, watershed segmentation, *etc.* [22]. However, morphological operation-based methods only deal with regular shapes, or where there are only bottleneck connections between touching objects.

A natural question is: *How does the human vision system parse a region into parts?* Perhaps such strategies can be adapted to solve our problem. Some theorists postulate that there is a set of basic shape primitives [3, 4, 18, 19] that are useful in finding parts and describing them. Others postulate that there are rules, based on geometric properties alone, by which humans perceive boundaries between parts for a given shape [10, 24]. Still others propose theories based on the relationship between the nonlinear diffusion equation and shape perception [14, 28]. We will make use of both shape constraints (deformable template and statistical priors) and geometric properties of the region contour (curvature minima) in guiding splits.

In related work, Hoffman and Richards [10, 20] provided the *minima rule* for shape parsing: cut each silhouette into parts at concave cusps and negative minima of curvature. For a 2D silhouette, the minima rule provides boundary points on the silhouette outline, and part cuts must pass through them. However, the minima rule does not define the part cuts themselves – it only constrains them to pass through the boundary points it provides.

There are perceptual constraints beyond the minima rule that affect one's parsing preference [24]. *Part salience* [9] and the *short-cut rule* [25] were proposed to embody those constraints. Hoffman and Singh[9] isolated three factors in part salience: relative area, amount of protrusion, and normalized curvature across the part boundary. However, they do not integrate these into a shape partitioning scheme. The short-cut rule [25] is simple: if boundary points can be joined in more than one way to parse a silhouette, choose the parsing that uses the shortest cuts. A cut is defined to be (1) a line that (2) crosses an axis of local symmetry, (3) joins two points on the silhouette, such that (4) at least one of the two points has negative curvature.

The short-cut rule has some limitations. First, the length

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
APR 2001		**00-00-2001 to 00-00-2001**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Region Segmentation via Deformable Model-Guided Split and Merge	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Office of Naval Research,One Liberty Center,875 North Randolph Street Suite 1425,Arlington,VA,22203-1995**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
The original document contains color images.

14. ABSTRACT
see report

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**		**7**	

of the cut only involves minimal shape information. Second, restricting the cut to cross a symmetry axis is problematic, because robust computation of symmetry axes is difficult in general. To circumvent these problems, [21] proposed segmenting regions into roughly convex parts via an optimization procedure at multiple scales. Since the method uses exhaustive search to find the optimal cuts, the algorithm is slow despite the use of a multi-scale approach. In addition, the method requires that the number of cuts for the shape partitioning is known.

Latecki, et al., [15] used a contour evolution method to identify convex parts at different stages. The assumption is that significant visual parts will become convex at higher stages of the evolution. In related work, Belyaev, et al.[1] developed a polygonal curve evolution method that can be used for multi-scale shape analysis. A major drawback of contour evolution approaches is that it is unclear how to distinguish cuts between objects from cuts between parts inside the same object.

Two-dimensional part decomposition methods to date have had several drawbacks. There is no cut rule that is good for all cases. However, our problem is different in that we can make use of a deformable template model in determining cuts employed in region splitting. Use of the shape model, which embodies knowledge about a particular class of objects and the prior distribution on nonrigid deformation, will improve cut selection considerably.

3 Review of Basic Approach

In [16, 23] we proposed a method that uses a deformable model to guide grouping of image regions. Our basic approach extends methods employed in active contours [13] and deformable templates [7, 8, 12, 17, 27] to the problem of automatic object detection and segmentation via region grouping. We will now briefly review our basic approach. For more detailed discussion of our approach, including parameter settings, please see [23].

An overview of the approach is shown in Fig. 1. In the pre-processing stage, the input image is over-segmented via standard region merging algorithms [2, 6]. The output of this module includes a standard region adjacency graph. An edge map is also computed; notable edges and their strengths are detected via standard image processing methods. The resulting edge map will be used to constrain consideration of possible grouping hypotheses later in region merging.

The system then tests various combinations of candidate region groupings to obtain an optimal labeling of the image. The shape model is deformed to match each grouping hypothesis $\mathbf{g}_i$ in such a way as to minimize a cost function:

$$E(\mathbf{g}_i) = \alpha E_{color} + (1-\alpha)((1-\beta)E_{area} + \beta E_{deform}), \quad (1)$$

where α and β are scalar constants with values in the range $[0, 1]$ that control the relative importance of the three terms:

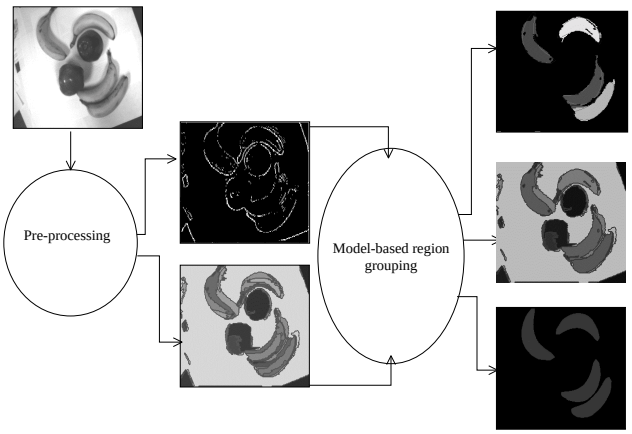

Figure 1: Diagram of the deformable template-based region merging system. An input image (image of bananas) undergoes pre-processing, which results in an over-segmentation and an edge map. These are inputs to the model-based region grouping stage (using a deformable banana template). The final output includes region groupings (shown in top output image) for detected objects (four bananas), and recovered models for the objects (shown in bottom output image).

E_{color} is a region color compatibility term for the region grouping, E_{area} is a region/model area overlap term, and E_{deform} is a deformation energy for the shape model. A model fitting procedure is used to compute the cost in Eq. 1 via the downhill simplex method.

A deformation term enforces *a priori* constraints on the amounts and types of deformations allowed for the template; *i.e.:*

$$E_{deform} \propto -\log P(\mathbf{a}|\Omega), \quad (2)$$

where $P(\mathbf{a}|\Omega)$ gives the prior distribution on global deformation parameters, $\mathbf{a}$, for a particular shape class Ω. In our experience, the use of a Gaussian model for the prior distribution on global deformation leads to reliable shape-based image segmentation. An estimate of the prior distribution is computed in a supervised fashion, for a given set of training examples for that shape class. In our implementation, linear and quadratic polynomials are used to model deformation due to stretching, shearing, bending, and tapering.

Further, in order to test the quality of a possible partitioning, a global cost function for partitioning the whole image is defined:

$$\mathcal{E} = (1-\gamma)\sum_{i=1}^{n} r_i \mathbf{E}(\mathbf{g}_i) + \gamma \mathbf{n}, \quad (3)$$

where γ is a constant factor, $\mathbf{n}$ is the number of the groupings in the current image partitioning, r_i is the ratio of i^{th} group area to the total area of connected regions, and $\mathbf{E}(\mathbf{g}_i)$ is the cost function for the group $\mathbf{g}_i$ (Eq. 1). The highest confidence first (HCF) algorithm [5] is used to find an approximately optimal value for Eq. 3. The approximately

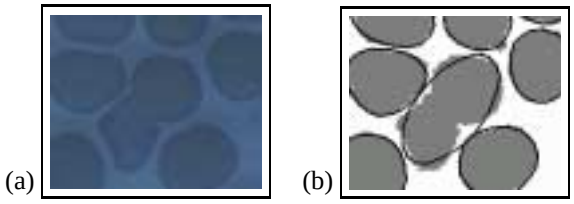

Figure 2: Error in segmentation The input image is a blood cell micrograph (a). The model-guided segmentation is shown in (b), with cell regions (grey areas) and fitting models (black contours).

optimal region groupings obtained via HCF along with recovered shape models are shown in Fig. 1.

4 Model-Guided Region Splitting

As observed in experiments [16], our model-guided region merging scheme is often able to obtain a satisfactory segmentation despite clutter, shadows, variation in illuminant, and shape deformation. However, performance of the method is hindered by its assumption of an over-segmented image to be used as input to the shape-based region grouping subsystem. In general, it is not possible to consistently over-segment all input images. Invariably, some parts of images are under-segmented, which means that some of the initial regions correspond with parts of more than one object in the scene. An example error in segmentation of a blood cell micrograph is shown in Fig. 2. In addition, the HCF optimization procedure sometimes becomes trapped in a local minimum that yields an incorrect region merging.

To address these problems, a region splitting step must be added to our system. Perceptually-motivated criteria will be used to determine where/how to split a region, based on the local shape properties of the region's bounding contour. We will also make use of our deformable shape-model in guiding the selection of splits. In considering a split, the change in the fitting cost (Eq. 3) after splitting provides a measure for the quality of the cut.

4.1 Region Splitting Criteria

In order to detect candidate regions or groups for splitting, the following test can be utilized:

1. At the beginning of the model-based region grouping stage, check each region. If that region's model fitting cost (Eq. 1) is larger than a threshold, then it is a splitting candidate.

2. At the end of merging via HCF, for each group, if its model fitting cost is larger than the threshold, then it is a splitting candidate.

The split threshold can be obtained through statistical analysis for the training examples. Since the number of regions in the over-segmentation result is large typically, we only utilize the splitting operation after the model-based HCF merging stage (for computational efficiency). If new regions occur in the model based splitting stage, then the model-based merging stage is invoked again to refine the final result. Eq. 3 is used both in splitting and merging for evaluating the feasibility of split/merge operations, enforcning the Minimum Description Length (MDL) principle.

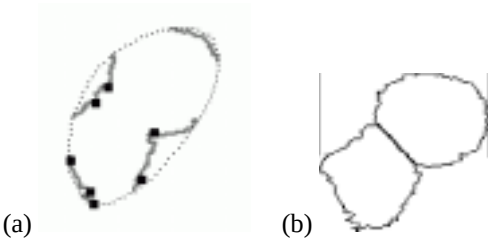

Figure 3: Detected cut points (a) and selected cut (b) for Fig. 2. Dotted contours in (a) are the model boundaries, dark square marks represents the candidate cut points.

4.2 Selecting Candidates of Cut Endpoints

If a group of regions is a splitting candidate, then we use perceptual criteria and model fitting information to select the splitting boundaries. After we have determined the overlapping between the fitted model and the underlying region for splitting, as shown in Fig. 2(b), we can get the curve segments of the region boundary inside the model area (the thick gray curve segments shown in Fig. 3(a)). The search for candidates of cut endpoints will be limited to these segments.

One criterion of splitting methods is robustness: the method should be insensitive to small changes of shape due to differences among the members of a class or to small differences in view position. Because curvature computation is more complex and sensitive to noise, we do not utilize the minima rule for the region splitting, instead we will search for cut endpoints in the concave segments on the region boundary along lines of [26]. Cut endpoints are generally located in the region boundary's concave segments, which are most likely inside the fitted model. Making use of the difference between the region boundary and model boundary can make the splitting algorithm more robust and more efficient since knowledge about the object can be employed in the splitting. The concave segments detection method is as follows:

1. Intersect the region boundary with the model boundary. This yields a collection of curve segments s_i of the region boundary that lie with in the model area.

2. For each curve segment s_i, for s_i's starting point to its end point, check the convexity of each point along the segment. In other words, for each point, check its small neighborhood to decide whether this point is on a convex segment or concave segment. The neighborhood size l_n

will influence the accuracy and robustness of the result.[1]

3. After smoothing the convexity of each point to eliminate noise influence, the whole segment s_i will be decomposed into concave and convex sub-segments.

4. For each concave sub-segment from s_i, check its length. If the length is compatible with the specified neighborhood size l_n for checking convexity, then select the center point of this sub-segment as a candidate cut point.

In the concave segments detection method, there is also another criterion which can be used in selecting candidate points[26]: If the length of a concave sub-segment is comparable with its adjacent convex sub-segments, then regard this convexity change as due to the natural shape change of the object, and not well suited for splitting. Only when the length of a concave sub-segment is significantly smaller than those of its adjacent convex sub-segments, is a candidate point selected from it. However, in our experiments, this criterion is too strict; real cut endpoints may be omitted, due to the non-smoothness of the region boundaries.

One added benefit in our approach is that we can make use of overlap between the deformed model and the underlying region for determining if and how to split the region. This makes the splitting algorithm more robust and efficient since knowledge about the object can be employed in testing and selecting cuts. It should be noted that our algorithm also considers the boundaries of holes in determining cut points, if the holes are inside the fitted model.

The cut point candidates obtained will be tested in the cut selection stage. It is likely that not all the cut point candidates will be used in the final cut (maybe none at all). Selection of cuts is described in the next section.

4.3 Strategies for Selecting a Cut

Given a set of candidate cut end points for splitting the region, there are many possible region cuts possible. In our model-based splitting method, the quality of a cut can be determined by calculating the change in global cost Eq. 3 that results from that region cut. Put differently, if the combined cost of the two new regions is less than the cost of the original region, then this cut can be regarded as feasible. For example, given region grouping $\mathbf{g}$ can be split into $\mathbf{g}_1$ and $\mathbf{g}_2$ by a cut, we compare the old cost

$$(1-\gamma)r\mathbf{E}(\mathbf{g})+\gamma, \tag{4}$$

and the new cost

$$(1-\gamma)(r_1\mathbf{E}(\mathbf{g}_1)+r_2\mathbf{E}(\mathbf{g}_2))+2\gamma, \tag{5}$$

where γ and r are as defined in Eq. 3, $\mathbf{E}(\mathbf{g})$ in Eq. 1. If the new cost is less than the old cost, then the cut is feasible.

To select a cut based on a set of candidates of cut endpoints, the direct strategy is: test all possible cuts. For all possible cuts, compare the cut quality via Eqs. 4 and 5 and select the one with the least cost. Fig. 3(b) shows a cut selected based on this strategy. The drawback of exhaustive search is the computation required. We therefore tested two alternative strategies for selecting a cut: *modified short-cut rule* [25], and a *queuing strategy*.

In the *modified short-cut rule*, the cut of shortest length is tested first. If this cut is a feasible cut, then stop. Otherwise, delete the candidate point with least significance from this cut, and repeat to test the shortest cuts given the remaining candidate points until a feasible cut is found or there is no cut remaining to test.

In the *queuing strategy*, sort all possible cuts according to the cut length and construct a queue. Fetch cuts from this queue until a feasible cut is obtained. In other words, first test the minimum cut, then the second minimum cut, *etc.*, until a feasible cut is found or no cuts remain.

If there is only one candidate point detected in a region for splitting, one possible solution is to generate a cut from this candidate point to cross the local symmetry axis of the region[25]. However, the robust computation of axes is difficult. Our simple alternative is to fit a line to the corresponding curve segment of the model boundary, and generate a cut from this candidate point to follow the normal direction of the fitted line. The motivation for this approximation is that model boundaries tend to be locally smooth.

4.4 Smooth Cuts from Model Boundaries

There is another benefit from computing the overlap between the recovered model and the underlying region. If one object is incorrectly merged with a small part of its adjacent object(as shown in Fig. 4(c) and (d)), the curve segments of model boundary inside the region are also good candidates of cuts since the recovered model is more stable to noise. In Fig. 4(c) and (d), one curve segment of the model boundary inside one fish region separates the incorrect merging part from the fish body. Based on curve segments of the model boundary inside the region, a smooth cut can be obtained instead of only straight cuts. Therefore, the precision of the region boundary can be improved as well. The resulting segmentation after split/merge step is shown in Fig. 4(e) and (f).

To reduce the computation complexity, smooth cuts are only considered if there is no feasible cut obtained through the strategies in Sec. 4.3 since the smooth cut strategy is more sensitive to small shape changes.

4.5 Recursive Splitting and Final Merging

Since we do not know how many objects exist in the region for splitting, we propose a recursive algorithm for splitting the region: after one feasible cut is found for the region, for the two new regions obtained via this splitting, recursively apply the splitting algorithm until no new feasible cut can be found. Therefore, multiple cuts can be found for one initial region. The recursion stops when no feasible cut

[1] We choose $l_n = 6$ in our experiments.

is found based on the strategies in Sec. 4.3 and Sec. 4.4. In order to refine the splitting result, we apply the HCF algorithm again to re-merge regions as needed.

5 Experiments

Experiments were conducted to evaluate the performance of the previous merge-only approach [16] versus the split-merge approach described in this paper. In the statistical analysis, we used a test database of 22 leaf images (about 200 leaf objects), 20 synthesized fish images (160 fish objects), and 21 cell micrographs (about 700 cell objects). Examples of these images are shown in Figs. 4– 6. Deformable template models were trained for each of the three shape classes using additional images that were not included in the test database as described in [16]. Between 40 and 70 images were used to train each shape model.

The synthetic images of fish were obtained as follows. Random fish shapes were created by drawing samples from fish template's deformation distribution (estimated during training). The resulting region was then partitioned into sub-regions via triangulation and assigned random colors drawn from a normal distribution. In addition, we added random noise to the synthesized images. Example synthetic fish images are shown in Fig. 4. Fish were placed at random orientations and positions in each image, subject to the constraint that they overlapped or touched at least one other fish.

Using the three databases, we conducted experiments to measure the success rate of object detection for the previous method (that used merge only) versus the new split/merge approach. The results of these experiments are shown in Table 1. Object detection rates were measured for all three cut selection strategies described in Sec. 4.3: short-cut rule, exhaustively test all cuts, and the queuing strategy. It was noted that there was a significant improvement in object detection rates when any of the splitting strategies was employed in concert with merging. However, there was no significant difference in object detection rates among the three different cut selection strategies. It was noted that the exhaustive, test all cuts strategy may not yield the globally optimal partioning always. This is evidenced by performance shown for the cell images in Table 1. At each step in this strategy, the locally optimal cut is selected from multiple cuts.

In the same experiments, it was also noted that segmentation quality was improved when split and merge were employed. As a quantitative measure for comparing the quality of segmentation, the model fitting cost was calculated for each region group using Eq.1. Table 2 gives a comparison of the mean fitting cost for deformed shape models recovered in the three different sets of test imagery. Smaller mean fitting costs generally indicate better fitting models, and higher-quality segmentation. Example results for merging versus split/merge are shown in Figs. 4–6.

	Leaves	Cells	Fish
No splitting	85.24%	85.79%	76.25%
Short-cut rule	92.86%	93.90%	85.00%
Test all cuts	96.19%	93.62%	86.25%
Queuing strategy	91.43%	93.90%	85.00%

Table 1: Comparison of object detection rates in three test sets.

	Leaves	Cells	Fish
No splitting	1.1338	1.1863	1.0626
Short-cut rule	1.1188	1.0531	1.0528
Test all cuts	1.1117	1.0522	1.0550
Queuing strategy	1.1148	1.0522	1.0555

Table 2: Comparison of the mean fitting cost for deformed shape models recovered in segmentation.

Given that there was not significant difference in performance between the three splitting strategies, a deciding factor in selecting a strategy would be CPU time required. In the experiments it was noted that on average the short-cut strategy was fastest; on average, obtaining the segmentation via short-cut strategy required only 59.6% of the CPU time required for exhaustive search. On average, the queuing strategy required 79.1% of the CPU time required for exhaustive search.

One limit of the system is that it cannot cope with large occlusions. An example is shown in the second row of Fig. 6, where two adjacent cells are not split due to a high degree of overlap. Our current algorithm does not explicitly model occlusion. It processes occluded objects as shapes with much more deformation. Incorporation of an explicit occlusion model is needed to deal with large overlapping of shapes.

6 Conclusion

An improved method for deformable shape-based image segmentation was described. The method employs a perceptually-motivated splitting strategy in concert with a deformable template model that embodies prior knowledge for a particular shape class. The method combines model fitting information in selecting cut points and also in the evaluation of the possible cuts in the region splitting. The recursive splitting structure in the algorithm gets rid of the requirement to know the number of cuts. This is an improvement over many previous shape splitting methods and multi-scale techniques[24, 15, 1, 21]. The experimental results show that after the splitting, the success rate of object detection and the accuracy of object boundaries are improved. Some different strategies are tested in our experiments for the feasible cut selection. When considering CPU time employed, the modified short-cut rule is faster in searching feasible cuts, and the splitting quality is comparable to other strategies.

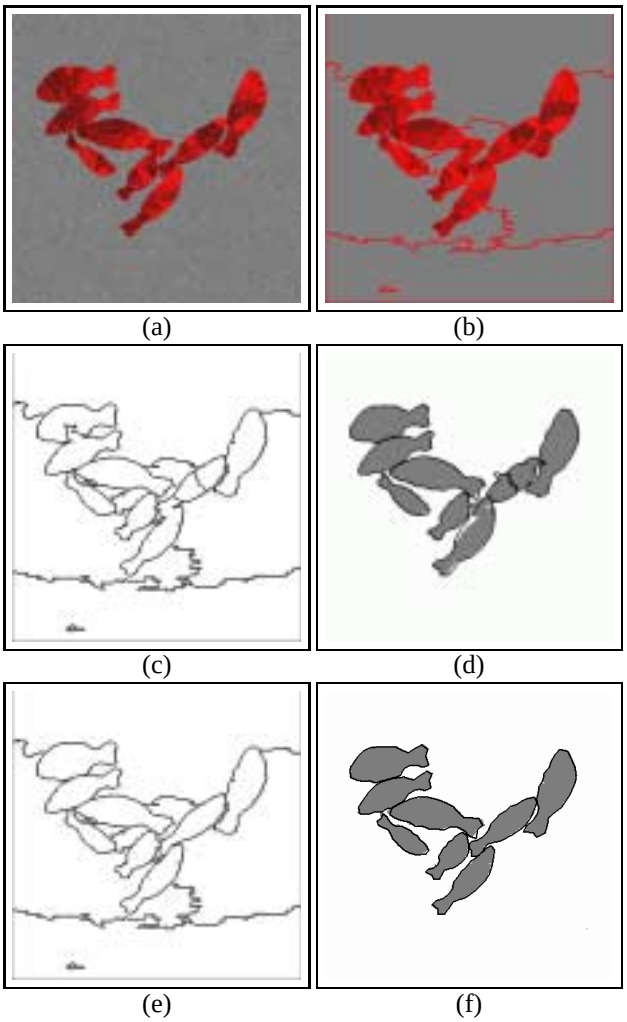

Figure 4: Region merging and split/merge experiment with a synthesized fish image. (a) Original image, (b) initial over-segmentation result, (c) merging result based HCF algorithm, (d) overlapping between the recovered models (thick black contours) and the regions (grey areas), (e) result of merging again after splitting, (f) new recovered models.

References

[1] A. Belyaev, et al. Polygonal curve evolutions for planar shape modeling and analysis. *Int. J. of Shape Modeling*, 5(2):195–217, 1999.

[2] J.R. Beveridge, et al. Segmenting images using localized histograms and region merging. *IJCV*, 2(3):311–352, 1989.

[3] I. Biederman. Recognition by components: A theory of human image understanding. *Psych. Review*, 94(2):115–147, 1987.

[4] T.O. Binford. Visual perception by computer. *IEEE Conf. on Sys. and Control*, 1971.

[5] P. B. Chou and C. M. Brown. The theory and practice of Bayesian image labeling. *IJCV*, 4(3):185–210, 1990.

[6] D. Comaniciu and P. Meer. Robust analysis of feature spaces: Color image segmentation. *CVPR*, 750–755, 1997.

[7] T.F. Cootes, A. Hill, C.J. Taylor, and J. Haslam. Use of active shape models for locating structure in medical images. *IVC*, 12(6):355–365, 1994.

[8] U. Grenander, Y. Chow, and D. M. Keenan. *Hands: A pattern theoretic study of biological shapes*. Springer-Verlag, 1991.

[9] D. D. Hoffman and M. Singh. Salience of visual parts. *Cognition*, 63(1):29–78, 1997.

[10] D.D. Hoffman and W.A. Richards. Parts of recognition. *Cognition*, 18:65–96, 1985.

[11] S.L. Horowitz and T. Pavlidis. A graph-theoretic approach to picture processing. *CGIP*, 7(2):282–291, 1978.

[12] A.K. Jain, Y. Zhong, and S. Lakshmanan. Object matching using deformable templates. *PAMI*, 18(3):267–278, 1996.

[13] M. Kass, A.P. Witkin, and D. Terzopoulos. Snakes: Active contour models. *IJCV*, 1(4):321–331, 1988.

[14] B.B. Kimia, A.R. Tannenbaum, and S.W. Zucker. Shapes, shocks, and deformations i: The components of 2-dimensional shape and the reaction-diffusion space. *IJCV*, 15(3):189–224, 1995.

[15] L. J. Latecki and R. Lakamper. Convexity rule for shape decomposition based on discrete contour evolution. *CVIU*, 73(3):441–454, 1999.

[16] L. Liu and S. Sclaroff. Deformable shape detection and description via model-based region grouping. *CVPR*, II:21–27, 1999.

[17] K. Mardia and K. de Souza. Deformable template recognition of multiple occluded objects. *PAMI*, 19(9):1035–1042, 1997.

[18] D. Marr. Analysis of occluding contour. *Proc. Royal Soc. B*, 197:441–475, 1977.

[19] A. Pentland. Automatic extraction of deformable part models. *IJCV*, 4(2):107–126, 1990.

[20] W. Richards, B. Dawson, and D. Whittington. Encoding contour shape by curvature extrema. *JOSA-A*, 3(9):1483–1491, 1986.

[21] P. L. Rosin. Shape partitioning by convexity. *IEEE Trans. on Sys. Man and Cyb. Part A* 30(2):202–210, 2000.

[22] J.C. Russ. *The Image Proc. Handbook*. CRC Press, 1992.

[23] S. Sclaroff and L. Liu. Deformable shape detection and description via model-based region grouping. *PAMI*, 23(5), 2001.

[24] M. Singh and B. Landau. Parts of visual shape as primitives for categorization. *Behavioral and Brain Sci.*, 21(1):36–37, 1998.

[25] M. Singh, G. D. Seyranian, and D. D. Hoffman. Parsing silhouettes: The short-cut rule. *Perception and Psychophysics*, 61(4):636–660, 1999.

[26] L. M. Vaina and S. D. Zlateva. The largest convex patches - a boundary-based method for obtaining object parts. *Biological Cybernetics*, 62(3):225–236, 1990.

[27] A. Yuille, D.S. Cohen, and P. Hallinan. Feature extraction from faces using deformable templates. *IJCV*, 8(2):99–111, 1992.

[28] S.C. Zhu and A. Yuille. Region competition: Unifying snakes, region growing, and Bayes/MDL for multiband image segmentation. *PAMI*, 18(9):884–900, 1996.

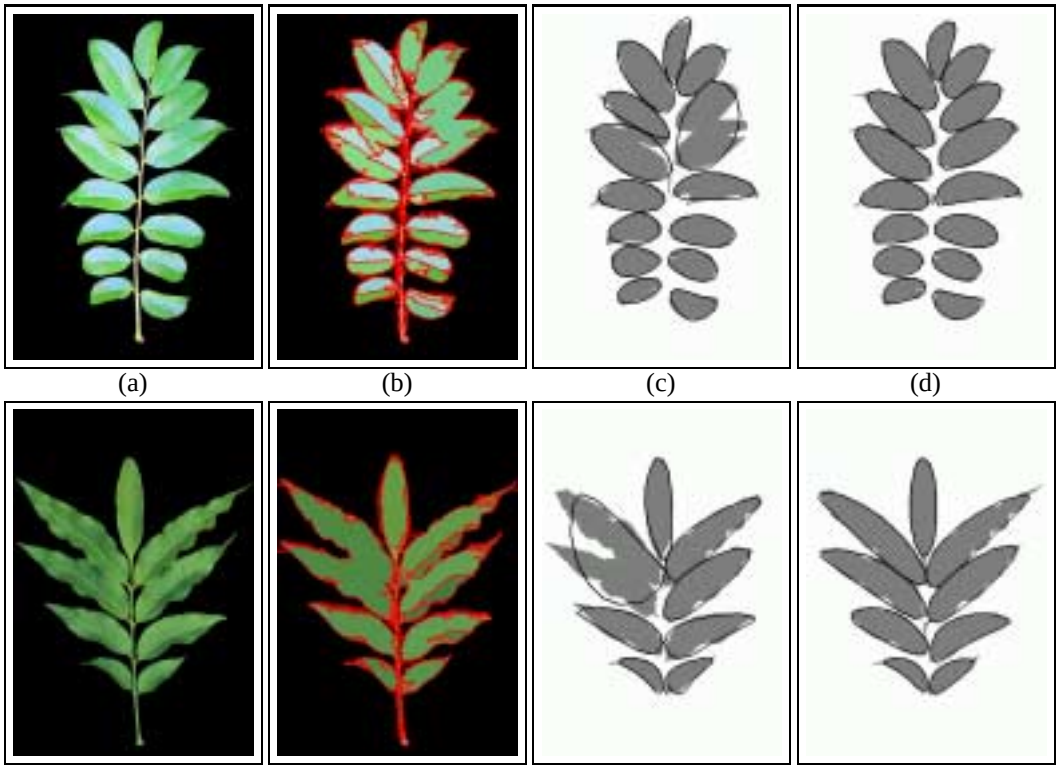

Figure 5: Two examples from the leaf image database: (a) original image, (b) over-segmentation, (c) merging result after HCF algorithm, showing overlap between the recovered models (thick black contours) and the regions (grey areas),(d) model-based splitting result.

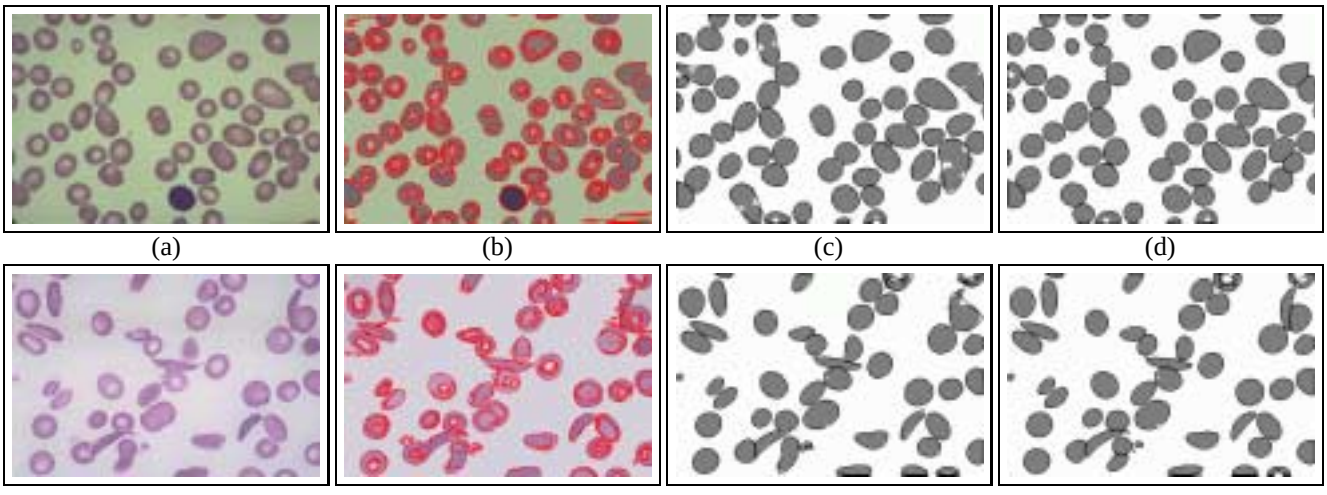

Figure 6: Example image from blood cell micrograph database: (a) original image, (b) over-segmentation result, (c) merging result based HCF algorithm showing overlap between the recovered models (thick black contours) and the regions (grey areas), (d) model-based splitting result.